



Scientific Working Group on Digital Evidence

SWGDE Technical Notes on FFmpeg

Disclaimer:

As a condition to the use of this document and the information contained therein, the SWGDE requests notification by e-mail before or contemporaneous to the introduction of this document, or any portion thereof, as a marked exhibit offered for or moved into evidence in any judicial, administrative, legislative or adjudicatory hearing or other proceeding (including discovery proceedings) in the United States or any Foreign country. Such notification shall include: 1) The formal name of the proceeding, including docket number or similar identifier; 2) the name and location of the body conducting the hearing or proceeding; 3) subsequent to the use of this document in a formal proceeding please notify SWGDE as to its use and outcome; 4) the name, mailing address (if available) and contact information of the party offering or moving the document into evidence. Notifications should be sent to secretary@swgde.org.

It is the reader's responsibility to ensure they have the most current version of this document. It is recommended that previous versions be archived.

Redistribution Policy:

SWGDE grants permission for redistribution and use of all publicly posted documents created by SWGDE, provided that the following conditions are met:

1. Redistribution of documents or parts of documents must retain the SWGDE cover page containing the disclaimer.
2. Neither the name of SWGDE nor the names of contributors may be used to endorse or promote products derived from its documents.
3. Any reference or quote from a SWGDE document must include the version number (or create date) of the document and mention if the document is in a draft status.

Requests for Modification:

SWGDE encourages stakeholder participation in the preparation of documents. Suggestions for modifications are welcome and must be forwarded to the Secretary in writing at secretary@swgde.org. The following information is required as a part of the response:

- a) Submitter's name
- b) Affiliation (agency/organization)
- c) Address
- d) Telephone number and email address
- e) Document title and version number
- f) Change from (note document section number)
- g) Change to (provide suggested text where appropriate; comments not including suggested text will not be considered)
- h) Basis for change

SWGDE Technical Notes on FFmpeg

Version: 1.0 (July 31, 2017)

This document includes a cover page with the SWGDE disclaimer.

Page 1 of 20



Scientific Working Group on Digital Evidence

Intellectual Property:

Unauthorized use of the SWGDE logo or documents without written permission from SWGDE is a violation of our intellectual property rights.

Individuals may not misstate and/or over represent duties and responsibilities of SWGDE work. This includes claiming oneself as a contributing member without actively participating in SWGDE meetings; claiming oneself as an officer of SWGDE without serving as such; claiming sole authorship of a document; use the SWGDE logo on any material and/or curriculum vitae.

Any mention of specific products within SWGDE documents is for informational purposes only; it does not imply a recommendation or endorsement by SWGDE.



Scientific Working Group on Digital Evidence

SWGDE Technical Notes on FFmpeg

Table of Contents

1. Purpose.....	4
2. Scope.....	4
3. Limitations	4
4. FFmpeg Tools	4
4.1 FFprobe	4
4.2 FFplay.....	4
4.3 FFmpeg.....	4
5. FFmpeg Installation	5
5.1 Windows Installation.....	5
5.2 Mac Installation.....	5
6. FFmpeg Informational Commands.....	6
6.1 Help	6
6.2 Show License	6
6.3 Installed Codecs	6
7. Basic Command Entry Format.....	6
7.1 FFprobe	6
7.2 FFplay.....	8
7.3 FFMPEG	10
7.4 Commonly Used FFMPEG Audio Commands	16
8. Video Analysis Commands.....	17
8.1 Create an MP4 Video with a Visual Display of Macroblock Types	17
8.2 Macroblock Analysis Visual Display Reference Chart.....	17
8.3 Video Frame Information Report.....	17
8.4 Frame-level MD5	18



Scientific Working Group on Digital Evidence

1. Purpose

This document provides a general awareness of FFmpeg, its functions, basic use, and common uses as it pertains to digital forensics. FFmpeg (Fast Forward mpeg) is an open source, cross-platform framework that uses command line to play, convert, and stream audio and video. This framework is used by multiple applications for forensic and commercial purposes.

2. Scope

The intended audience is forensic video analysts/examiners trained and experienced in the examination of video seeking direction and familiarization in the use of FFmpeg's open source suite. It focuses on basic commands and principles, as well as some commands commonly used in video analysis. The nature of FFmpeg is that it is constantly being collaborated and expanded. As more commands prove useful, they can be added to this document.

3. Limitations

This document was prepared with the resources available at the time of publication. As with all technology, FFmpeg is a constantly evolving environment with frequent implementation of new features and innovations. The specific configuration of any particular installation will vary widely and may not conform to the standards cited here. This document is not intended for use as a step-by-step guide for conducting a thorough forensic investigation, nor is it legal advice. While FFmpeg will process many video codecs, it may not work in every instance. This is not a best practices guideline and should not replace proper training and tool validation.

4. FFmpeg Tools

4.1 FFprobe

A multimedia tool that provides metadata about digital multimedia including but not limited to duration, frame rate, frame size, aspect ratio, codec, streams (video, audio, and data), etc.

4.2 FFplay

A media player that utilizes the FFmpeg framework to display multimedia files. While not all encompassing, it will play proprietary and open source media files.

4.3 FFmpeg

A command line tool to convert multimedia file formats as well as various properties within the file such as codecs, container, frame rate, aspect ratio, and still images.



Scientific Working Group on Digital Evidence

5. FFmpeg Installation

5.1 Windows Installation

- Download the latest STATIC version from ffmpeg.zeranoe.com/builds/
- Use 7zip compatible software to "extract all" from the downloaded file; it will put everything in one folder
- On the C drive, create a folder named "ffmpeg"
- Copy everything from the extracted folder into the "ffmpeg" folder on the C drive
- Right click on the Windows button. Select "Control Panel", click "System and Security", click "System", click "Advanced System Settings"
- Click on "Environment Variables"
- Under "User Variables for User", click "New"
- Under "Variable Name" type the word "Path"
- Under "Variable Value" type "c:\ffmpeg\bin"
- Click "Ok"

5.2 Mac Installation

Homebrew is a command-line package manager that will automatically install and attach dependencies. Using Homebrew requires both an internet connection and installation of Homebrew first. The steps below will address installation of Homebrew and FFmpeg.

- In Terminal, type: `/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`
- Follow the on-screen instructions; this will take a few minutes while it's installing the necessary developer tools for OS X
- After successful installation run: `brew install ffmpeg` to get the latest released version and library dependencies
- To see additional installation options, run `brew info ffmpeg`. These may include:

```
brew install ffmpeg
  • --with-fdk-aac
  • --with-ffplay
  • --with-freetype
  • --with-libass
  • --with-libquvi
  • --with-libvorbis
  • --with-libvpx
  • --with-opus
  • --with-x265
```
- To update ffmpeg, run: `brew update && brew upgrade ffmpeg`

Note: For these instructions and additional options visit:

<https://trac.ffmpeg.org/wiki/CompilationGuide/MacOSX>



Scientific Working Group on Digital Evidence

6. FFmpeg Informational Commands

6.1 Help

FFprobe -h
FFplay -h
FFmpeg -h

6.2 Show License

(lower-case L)
FFprobe -l
FFplay -l
FFmpeg -l

```
ffmpeg version N-85266-g1229007 Copyright (c) 2000-2017 the FFmpeg developers
  built with gcc 6.3.0 (GCC)
  configuration: --enable-gpl --enable-version3 --enable-cuda --enable-cuvid --enable-d3d11va --enable-dxva
  pmfx --enable-nvenc --enable-avisynth --enable-bzlib --enable-fontconfig --enable-frei0r --enable-gnutls --
  enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libfreetype --enable-libgme --
  enable-liblilbc --enable-libltdl --enable-libmodplug --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-
  libopenh264 --enable-libopenjpeg --enable-libopus --enable-librtmp --enable-libsnpappy --enable-libsoxr --
  enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvo-amrwbenc --enable-libvorbis --
  enable-libvpx --enable-libwebp --enable-libx264 --enable-libx265 --enable-libxavs --enable-libxvid --
  enable-lzma --enable-zlib
  libavutil      55. 59.100 / 55. 59.100
  libavcodec     57. 90.100 / 57. 90.100
  libavformat    57. 72.100 / 57. 72.100
  libavdevice    57.  7.100 / 57.  7.100
  libavfilter    6. 83.100 / 6. 83.100
  libswscale     4.  7.100 / 4.  7.100
  libswresample  2.  8.100 / 2.  8.100
  libpostproc   54.  6.100 / 54.  6.100
  hyper fast Audio and Video encoder
  usage: ffmpeg [options] [[infile options] -i infile]... {[outfile options] outfile}...
```

6.3 Installed Codecs

FFprobe -codecs
FFplay -codecs
FFmpeg -codecs

```
codecs:
D..... = Decoding supported
E..... = Encoding supported
..V... = Video codec
..A... = Audio codec
..S... = Subtitle codec
...I.. = Intra frame-only codec
....L = Lossy compression
.....S = Lossless compression
-----
D.VI.. 012v      Uncompressed 4:2:2 10-bit
D.VI.. 4xm      4X Movie
D.VI..S 8bps    QuickTime 8BPS video
E.VI.L a64_multi1 Multicolor charset for Commodore 64 (encoders: a64multi1)
E.VI.L a64_multi5 Multicolor charset for Commodore 64, extended with 5th color (colram) (encoders: a64multi5)
)
D.V..S aasc     Autodesk RLE
D.VI.L aic      Apple Intermediate Codec
DEVI..S alias_pix Alias/Wavefront PIX image
DEVI.L amv      AMV Video
D.V..L amn      Deluxe Paint Animation
D.V..L ansi     ASCII/ANSI art
DEVI..S apng    APNG (Animated Portable Network Graphics) image
DEVI.L asv1     ASUS V1
DEVI.L asv2     ASUS V2
D.VI.L aura     Auravision AURA
D.VI.L aura2    Auravision Aura 2
D.V..L av1      Alliance for Open Media AV1
D.V... avrn     Avid AV1 Codec
DEVI.. avrp     Avid 1:1 10-bit RGB Packer
```

7. Basic Command Entry Format

In this document, all input files will be referred to as “input.dvr”.

7.1 FFprobe

```
ffprobe input.dvr
```

FFprobe

Starts the command

input.dvr

Gives the location of the file

Enter

Runs the command

Shortcut: drag the file into the FFmpeg window to populate the path address into the command line.



Scientific Working Group on Digital Evidence

```
Command Prompt
C:\WINDOWS\system32>ffprobe "D:\Photo Flight\00254.MTS"
ffprobe version N-85266-g1229007 Copyright (c) 2007-2017 the FFmpeg developers
  built with gcc 6.3.0 (GCC)
  configuration: --enable-gpl --enable-version3 --enable-cuda --enable-cuvid --enable-d3d11va --enable-dxva2 --enable-lib
  bmf-x --enable-nvenc --enable-avisynth --enable-bzlib --enable-fontconfig --enable-frei0r --enable-gnutls --enable-iconv
  --enable-libass --enable-libbluray --enable-libs2b --enable-libcaca --enable-libfreetype --enable-libgme --enable-libgs
  m --enable-libilbc --enable-libmodplug --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb --enabl
  e-libopenh264 --enable-libopenjpeg --enable-libopus --enable-librtmp --enable-libsndio --enable-libsoxr --enable-libspe
  ex --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvo-amrwbenc --enable-libvorbis --enable-libvpx
  --enable-libwavpack --enable-libwebp --enable-libx264 --enable-libx265 --enable-libxavs --enable-libxvid --enable-libzi
  mg --enable-lzma --enable-zlib
  libavutil      55. 59.100 / 55. 59.100
  libavcodec     57. 90.100 / 57. 90.100
  libavformat    57. 72.100 / 57. 72.100
  libavdevice    57.  7.100 / 57.  7.100
  libavfilter     6. 83.100 /  6. 83.100
  libswscale     4.  7.100 /  4.  7.100
  libswresample  2.  8.100 /  2.  8.100
  libpostproc   54.  6.100 / 54.  6.100
Input #0, mpegts, from 'D:\Photo Flight\00254.MTS':
  Duration: 00:00:43.46, start: 1.033367, bitrate: 23760 kb/s
  Program 1
    Stream #0:0[0x1011]: Video: h264 (High) (HDMV / 0x564D4448), yuv420p(top first), 1920x1080 [SAR 1:1 DAR 16:9], 29.97
    fps, 59.94 tbr, 90k tbn, 59.94 tbc
    Stream #0:1[0x1100]: Audio: ac3 (AC-3 / 0x332D4341), 48000 Hz, stereo, fltp, 256 kb/s
    Stream #0:2[0x1200]: Subtitle: hdmv_pgs_subtitle ([144][0][0][0] / 0x0090), 1920x1080
C:\WINDOWS\system32>
```

Sample result

7.1.1 Sample result from FFprobe:

Duration: 00:00:43.46, start: 1.033367, bitrate: 23760 kb/s

Stream #0:0[0x1011]: Video: h264, yuv420p(top first), 1920x1080 [SAR 1:1 DAR 16:9], 2500 kb/s, 29.97 fps, 59.94 tbr, 90k tbn, 59.94 tbc

Stream #0:1[0x1100]: Audio: ac3 (AC - 3 / 0x332D4341), 48000 Hz, stereo, fltp, 256 kb/s

Stream #0:2[0x1200]: Subtitle: hdmv_pgs_subtitle ([144][0][0][0] / 0x0090), 1920x1080

7.1.2 Explanation of results:

Duration

Shows the duration of the file: 00:00:43.46

Stream

Shows the number of streams in the file. This instance has 3 streams: stream #0:0 which is identified as video, stream #0:1 which is identified as audio, and stream #0:2 which is identified as subtitles

Video

Shows the video codec used in the file. This file is identified as using h264

1920x1080

Shows the pixel dimensions of the video

29.97 fps

Shows the frames per second (fps)

SWGDE Technical Notes on FFmpeg

Version: 1.0 (July 31, 2017)

This document includes a cover page with the SWGDE disclaimer.

Page 7 of 20



Scientific Working Group on Digital Evidence

FFprobe can be used to provide generic metadata information about the file or as complex as byte level analysis of individual frames, audio streams and data streams.

7.2 FFplay

```
ffplay input.dvr
```

FFplay

Starts the command

input.dvr

Gives the location and name of the video to be played

Enter

Runs the command



Video window opens and begins playback.

If video does not play, use the metadata provided by FFprobe to instruct FFplay to force the format of the video file. Example below is forcing h264 format.

```
ffplay -f h264 input.dvr
```

FFplay

Starts the command

-f h264

Forces the format to h264

input.dvr

Gives the location and name of the file



Scientific Working Group on Digital Evidence

Other commonly used formats include but are not limited to:

- H264
- M4V
- MJPEG
- MPEG2
- H263

For a list of other formats available in FFplay type:

FFplay -formats

FFplay controls for operation of playback window

- Pause/Play: P key or Spacebar
- Scrub forward: On the video window, click and drag mouse to the right
- Scrub backward: On the video window, click and drag mouse to the left
- Frame forward: S key
- Display audio waveform: W key
- Toggle view of waveform or spectral frequency display: W key
- Full screen: F key
- To quit or close window: Q key or ESC key
- To mute: M key
- Set Volume: Press and hold 9 key to lower volume or 0 key to increase volume
- Decrease/Increase volume: / and * respectively
- Cycle audio streams: A
- Cycle video streams: V
- Cycle subtitle streams: T
- Cycle program: C
- Seek backward/forward 10 seconds: Left/Right keys
- Seek backward/forward 1 minute: Down/Up keys
- Seek backward/forward 10 minutes: Page Down/Page Up



Scientific Working Group on Digital Evidence

7.3 FFMPEG

This is the basic command structure for FFmpeg; all other commands will follow this structure:

```
[Call Program][input Arguments]-i[Input File][Output Arguments][Output file]
```

For example, this command plays the inputted file at 10 frames per second, and transcodes it to an uncompressed AVI.

```
ffmpeg -r 10 -i input.dvr -c:v rawvideo -f avi output.avi
```

FFmpeg

Starts the command.

-r 10

Tells FFmpeg the video input should be played at 10 frames per second.

-i input.dvr

Path and name of input file

-c:v rawvideo

Tells FFmpeg to transcode the video stream to an uncompressed video format

-f avi

Tells FFmpeg to wrap the video in an AVI container with necessary header information

Output.avi

Path and name of output file

```
FF Prompt
nable-bzlib --enable-fontconfig --enable-frei0r --enable-gnutls --enable-iconv --enable-
libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcadec --enable-
libfreetype --enable-libgme --enable-libgsm --enable-libilbc --enable-libmodplug --enabl
e-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb --enable-libopenjpeg
--enable-libopus --enable-librtmp --enable-libschrödinger --enable-libsoxr --enable-lib
speex --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvo-aacenc -
enable-libvo-amrwbenc --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-l
ibwebp --enable-libx264 --enable-libx265 --enable-libxavs --enable-libxvid --enable-lzma
--enable-decklink --enable-zlib
libavutil      55.  3.100 / 55.  3.100
libavcodec     57.  5.100 / 57.  5.100
libavformat    57.  3.100 / 57.  3.100
libavdevice    57.  0.100 / 57.  0.100
libavfilter     6. 11.100 /  6. 11.100
libswscale     4.  0.100 /  4.  0.100
libswresample  2.  0.100 /  2.  0.100
libpostproc   54.  0.100 / 54.  0.100

For help run: ffmpeg -h
For formats run: ffmpeg -formats | more
For codecs run:  ffmpeg -codecs | more

Current directory is now: "C:\ffmpeg\bin"
The bin directory has been added to PATH

>Ffmpeg -i D:\myfile.mp4 -c:v rawvideo -f avi D:\myoutput.avi
```

Commonly Used FFmpeg Commands

SWGDE Technical Notes on FFmpeg

Version: 1.0 (July 31, 2017)

This document includes a cover page with the SWGDE disclaimer.



Scientific Working Group on Digital Evidence

7.3.1 To change the file container without transcoding the stream information:

```
ffmpeg -i input.dvr -c:v copy -f avi output.avi
```

FFmpeg

Starts the command

-i input.dvr

Path and name of input file

-c:v copy

Tells FFmpeg to copy the original video stream data

-f avi

Tells FFmpeg to wrap the video in an AVI container with necessary header information

output.avi

Path and name of output file

7.3.2 Output uncompressed AVI

```
ffmpeg -i input.dvr -c:v rawvideo -f avi output.avi
```

FFmpeg

Starts the command

-i input.dvr

Path and name of input file

-c:v rawvideo

Tells FFmpeg to transcode the video stream to an uncompressed video format

-f avi

Tells FFmpeg to wrap the video in an AVI container with necessary header information

output.avi

Path and name of output file



Scientific Working Group on Digital Evidence

7.3.3 Force FFmpeg to read file with a specific codec and change the file container

```
ffmpeg -f h264 -i input.dvr -c:v copy -f avi output.avi
```

FFmpeg

Starts the command

-f h264

Tells FFmpeg the video stream should be forced to play as h264, regardless of file data. Other common dvr codecs are mjpeg, h263, and m4v

-i input.dvr

Path and name of input file

-c:v copy

Tells FFmpeg to copy the inputted video codec

-f avi

Tells FFmpeg to wrap the video in an AVI container with necessary header information

Output.avi

Path and name of output file

7.3.4 Change the framerate and the file container

```
ffmpeg -r 10 -i input.dvr -c:v copy -f avi output.avi
```

FFmpeg

Starts the command

-r 10

Tells FFmpeg the video input should be played at 10 frames per second. This can come after the input file, but that is not forensically recommended, as that tells FFmpeg to play the output file at the listed frame rate, potentially adding frames to pad.

-i input.dvr

Path and name of input file

-c:v copy

Tells FFmpeg to copy the inputted video codec

-f avi

Tells FFmpeg to wrap the video in an AVI container with necessary header information

Output.avi

Path and name of output file



Scientific Working Group on Digital Evidence

7.3.5 Output frames from video

7.3.5.1 Output all video file I-frames to uncompressed sequential still images

```
ffmpeg -i input.dvr -vsync drop -vf select='eq(pict_type,I)' -f image2 -pix_fmt rgb24 foldername\frame%05d.tiff
```

FFmpeg

Starts the command

-i input.dvr

Path and name of input file

-vsync drop

Video sync option which tells FFmpeg to disregard timestamp information

-vf select='eq(pict_type,I)'

Tells FFmpeg to select frames based on a picture type, in this case I-frames. Other options include P for P-frames, and B for B-frames. Can be comma delimited to combine

-f image2

Tells FFmpeg to encode the selected image type(s) into an image sequence

-pix_fmt rgb24

Tells FFmpeg to format the images as 24 bit RGB images

foldername\frame%05d.tiff

Specifies an output folder and a title for the images numbered in a 5 digit sequence as TIFF files.

7.3.5.2 Output individual frames from video stream

```
ffmpeg -i input.[extension] -an -f image2 output_%05d.[extension]
```

FFmpeg

Starts the command

-i input.[extension]

Path and name of input file

-an

Tells FFmpeg to exclude any existing audio tracks in the file

-f image2

Tells FFmpeg to extract still images for each frame of video

output %05d.[extension]

Path and name of output file, the extension will force the still image extraction to a desired still image format, e.g., .jpg, .tiff, or .png.



Scientific Working Group on Digital Evidence

7.3.6 Export a portion of the video as an uncompressed avi

```
ffmpeg -ss 0:00:10.000 -i input.dvr -t 0:00:05.000 -c:v rawvideo -f avi  
output.avi
```

FFmpeg

Starts the command

-ss 0:00:10.000

Tells FFmpeg to seek to a time in the video using the format
hours:minutes:seconds.milliseconds

-i input.dvr

Path and name of input file

-t 0:00:05.000

Tells FFmpeg to play for a specific amount of time. Format is the same as -ss

-c:v rawvideo

Tells FFmpeg to transcode the video stream to an uncompressed video format

-f avi

Tells FFmpeg to wrap the video in an AVI container with necessary header
information

output.avi

Path and name of output file

7.3.7 Concatenate common files into an avi

There are some caveats with concatenation. Before concatenating, all videos should be from the same source, with the same frame rate and the same codec. The basic command for this is:

```
ffmpeg -i "concat:input1.dvr|input2.dvr|input3.dvr" -c copy output.dvr
```

FFmpeg

Starts the command

-i "concat:input1.dvr|input2.dvr|input3.dvr"

Tells FFmpeg to combine the files in the order they are entered. FFmpeg will start with the first file in the list, and continue until the last file is completed.

-c copy

Tells FFmpeg to copy the inputted codec. This will include all streams within the file (e.g., video, audio, timecode)

output.dvr

Path and name of output file



Scientific Working Group on Digital Evidence

For more efficient concatenation of multiple files, the input can be done from a list.txt file. The text in the list needs to be structured as follows:

file 'input1.dvr'

file 'input2.dvr'

file 'input3.dvr'

This can be created manually, or as a loop. For example, in Windows command-line, this command will create a list.txt file for all similar files in a folder.

```
(for %i in (*.dvr) do @echo file '%i') > list.txt
```

Once a list.txt file has been created, use the following command to combine all listed files into one file.

```
ffmpeg -f concat -i list.txt -c copy -f avi output.dvr
```

FFmpeg

Starts the command

-f concat

Tells FFmpeg to concatenate all the files in the input

-i list.txt

Is the path and name of the list.txt file. FFmpeg will start with the first file in the list and continue until the last file is completed.

-c copy

Tells FFmpeg to copy the inputted codec. This will include all streams within the file (e.g., video, audio, timecode)

-f avi

Tells FFmpeg to wrap the video in an AVI container with necessary header information

Output.avi

Path and name of output file

One note from FFmpeg documentation: If using MP4 files, these could be losslessly concatenated by first transcoding them to mpeg transport streams. With h.264 video and AAC audio, the following commands can be used:

```
ffmpeg -i input1.mp4 -c copy -bsf:v h264_mp4toannexb -f mpegts temp1.ts  
ffmpeg -i input2.mp4 -c copy -bsf:v h264_mp4toannexb -f mpegts temp2.ts  
ffmpeg -i "concat:temp1.ts|temp2.ts" -c copy -bsf:a aac_adtstoasc output.mp4
```



Scientific Working Group on Digital Evidence

7.4 Commonly Used FFMPEG Audio Commands

7.4.1 Output Audio Only from a File

```
ffmpeg -i input1.dvr -c:a copy -vn -f wav output.wav
```

FFmpeg

Starts the command

-i input.dvr

Path and name of input file

-c:a copy

Tells FFmpeg to copy the inputted audio codec

-vn

Tells FFmpeg to ignore and remove any video stream present

-f wav

Tells FFmpeg to wrap the audio in a wav container with necessary header information

output.wav

Path and name of output file

7.4.2 Output uncompressed AVI with lossless audio

```
ffmpeg -i input.dvr -c:a pcm_s16le -ar 44100 -c:v rawvideo -f avi output.avi
```

FFmpeg

Starts the command

-i input.dvr

Path and name of input file

-c:a pcm_s16le

Tells FFmpeg to transcode the audio stream to an uncompressed 16-bit audio format

-ar 44100

Tells FFmpeg to sample the transcoded audio as a Sample Rate of 44.1kHz

-c:v rawvideo

Tells FFmpeg to transcode the video stream to an uncompressed video format

-f avi

Tells FFmpeg to wrap the video in an AVI container with necessary header information

output.avi

Path and name of output file



Scientific Working Group on Digital Evidence

8. Video Analysis Commands

8.1 Create an MP4 Video with a Visual Display of Macroblock Types

(Note - Decoding of macroblock types is MPEG-specific)

```
ffmpeg -debug vis_mb_type -i input.dvr output.mp4
```

FFmpeg

Starts the command

-debug vis_mb_type

Tells FFmpeg to visualize block types

-i input.dvr








Path and name of input file

output.mp4

Path and name of output file (MP4 format)

8.2 Macroblock Analysis Visual Display Reference Chart

(Colors below are displayed in the order that they are parsed)

<u>Color</u>	<u>Reference</u>
	New data
	New data
	16x16 Skip macroblock (P or B slices)
	Reference to past (List 0, P or B slices)
	Reference to future (List 1, B slices)
	Reference to past and future (List 1 & 2, B slices)
	Generally indicates no change to the macroblock

8.3 Video Frame Information Report

This report shows time and display information for each frame in a video.

```
ffprobe -show_frames -print_format xml input.dvr > output.xml
```

FFprobe

Starts the command

-show_frames

Shows information for each frame

-print_format xml

Tells FFprobe to send frame information into an XML file

input.dvr>output.xml

Gives the name of the input file and path, and sends it to the listed output.xml name and path



Scientific Working Group on Digital Evidence

8.4 Frame-level MD5

This procedure provides a methodology for verifying lossless changing of a codec from one container to another container using ffmpeg's framemd5 command, which currently only supports the md5 algorithm. First, generate md5 values for every frame in a video file. Then, change the container of the video file into a new container (e.g., AVI, MOV). Then create md5 values for the new video file. Finally, compare the two sets of framemd5 values to prove that the video/audio information has not changed during the container changing process. The following sections provide instructions for this process.

8.4.1 Generate md5 values for every frame in a video file

First use ffmpeg to create a text file that includes an md5 value for each frame in the first file that you want to compare.

```
ffmpeg -i input.dvr -f framemd5 file1.md5.txt
```

FFMPEG

Starts the command

-i input.dvr

Path and name of the input file

-f framemd5

Instructs ffmpeg to create md5 hash values for each frame of video data

file1.md5.txt

Instructs ffmpeg to output the result of the framemd5 to a particular text file

8.4.2 Create md5 values for the new video file

Using a second file that you want to compare to the first file, create a text file that includes an md5 value for each frame in the second file.

```
ffmpeg -i input.dvr -f framemd5 file2.md5.txt
```

FFMPEG

Starts the command

-i file2

Path and name of input file

-f framemd5

Instructs ffmpeg to create md5 hash values for each frame of video data

file2.md5.txt

Instructs ffmpeg to output the result of the framemd5 to a particular text file



Scientific Working Group on Digital Evidence

8.4.3 Compare the two sets of framemd5 values

In order to compare the two files, use a comparison technique to compare the text of the two framemd5 files to look for any differences in them. If there are no differences, then you have an exact match in frame count, frame size, and byte order per frame.

```
FC file1.md5.txt file2.md5.txt > result.txt
```

FC

FC is a Windows operating system command to compare the textual content of two specified files

file1.md5.txt

Specifies the first input file

file2.md5.txt

Specifies the second input file

> result.txt

Instructs FC to output the result of the comparison to a particular text file



Scientific Working Group on Digital Evidence

SWGDE Technical Notes on FFmpeg

History

Revision	Issue Date	Section	History
1.0 DRAFT	2016-09-15	All	Initial drafted and SWGDE voted to release as a Draft for Public Comment.
1.0 DRAFT	2016-10-08	All	Formatted and technical edit performed for release as a Draft for Public Comment.
1.0 DRAFT	2017-01-12	7.3, 7.4, 8.4	Added Section 7.3.5.2 Output individual frames from video stream. Added Section 7.4 Commonly Used FFMPEG Audio Commands. Added 8.4 Frame-level MD5. SWGDE voted to re-release as a Draft for Public Comment.
1.0 DRAFT	2017-02-21	Formatting	Formatted and published as a Draft for Public Comment.
1.0 DRAFT	2017-05-30	All images	Updated all images with new, original images. Re-released as a Draft for Public Comment.
1.0	2017-06-22	None	SWGDE voted to publish as Approved, pending completion of 60-day draft period if no additional comments are received.
1.0	2017-07-31	Formatting	Formatting and technical edit performed and published as Approved version 1.0.